

## CLAIMS

What is claimed is:

1. An electronic system, comprising:  
a processor;  
a network interface controller including a hardware port;  
a virtual switch comprising software executed by said processor and including a plurality of virtual ports, said virtual ports adapted to provide communication between an application running on said processor and said network interface controller; and  
an application programming interface ("API") running on said processor and usable by said application to interface with said virtual switch.
2. The electronic system of claim 1 wherein said API includes code that permits an application to register itself with the virtual switch to permit a resource to be assigned to said application.
3. The electronic system of claim 1 wherein said API includes code that permits an application to register itself with the virtual switch to permit a unique identifier to be assigned to said application.
4. The electronic system of claim 1 wherein said API further includes code to deregister said application from virtual switch to release a resource that has been assigned for use by said application.
5. The electronic system of claim 1 wherein said API includes code to permit said application to transmit data through said virtual switch to another application.
6. The electronic system of claim 1 wherein said API includes code to cause said virtual switch to open a handle and to post a receive buffer on said handle.
7. The electronic system of claim 6 wherein said code that causes the virtual switch to open a handle and post a receive also includes code to transition said

handle between a first state and a second state, said first state indicates that the switch has not received data to be provided to said application and said second state indicates that the virtual switch has received data to be provided to said application.

8. The electronic system of claim 6 wherein said API also includes code for closing said handle.

9. The electronic system of claim 1 wherein said API includes to permit the application to inform the virtual switch that the application is ready to receive data.

10. The electronic system of claim 6 wherein said API includes to permit the application to inform the virtual switch that the application is ready to receive data.

11. A network, comprising:  
a plurality of nodes; and  
at least one switch coupling the nodes together;  
wherein each of said nodes includes:  
a processor;  
a network interface controller including a hardware port;  
a virtual switch comprising software executed by said processor  
and including a plurality of virtual ports, said virtual ports  
adapted to be provide communication between an applica-  
tion running on said processor and said network interface  
controller; and  
an application programming interface ("API") running on said  
processor and usable by said application to interface with  
said virtual switch.

12. The network of claim 11 wherein said API includes code that permits an application to register itself with the virtual switch to permit a resource to be assigned to said application.

13. The network of claim 11 wherein said API includes code that permits an application to register itself with the virtual switch to permit a unique identifier to be assigned to said application.

14. The network of claim 11 wherein said API further includes code to deregister said application from virtual switch to release a resource that has been assigned for use by said application.

15. The network of claim 11 wherein said API includes code to permit said application to transmit data through said virtual switch to another application.

16. The network of claim 11 wherein said API includes code to cause said virtual switch to open a handle and to post a receive buffer on said handle.

17. The network of claim 16 wherein said code that causes the virtual switch to open a handle and post a receive also includes code to transition said handle between a first state and a second state, said first state indicates that the switch has not received data to be provided to said application and said second state indicates that the virtual switch has received data to be provided to said application.

18. The network of claim 16 wherein said API also includes code for closing said handle.

19. The network of claim 1 wherein said API includes to permit the application to inform the virtual switch that the application is ready to receive data.

20. The network of claim 16 wherein said API includes to permit the application to inform the virtual switch that the application is ready to receive data.

21. A computer readable storage medium storing instructions that when executed by a processor cause the processor to implement an application

programming interface for a virtual switch, said virtual switch implemented in software on a computer, said instructions comprising:

- an instruction usable to allocate a resource to an application to permit said application to access said virtual switch;
- an instruction usable to permit said application to transmit data through said virtual switch; and
- an instruction usable to receive data provided to said application through said virtual switch.

22. The storage medium of claim 21 further including an instruction usable to deallocate said resource upon said application ceasing using said virtual switch.

23. The storage medium of claim 21 further including an instruction usable to cause said virtual switch to open a handle and to post a receive buffer on said handle.

24. The storage medium of claim 23 further including an instruction usable to close said handle.

25. A method, comprising:  
registering an application with a software implemented switch; and  
executing an instruction to cause said virtual switch to provide data to said application.

26. The method of claim 25 wherein registering includes providing a virtual port identifier to said application.

27. The method of claim 25 wherein registering includes allocating a resource to said application.

28. The method of claim 25 further including deregistering said application from said software-implemented switch.

29. The method of claim 25 further including providing an instruction that causes data to be transmitted from said application to said software-implemented switch.
30. An electronic system, comprising:  
a processor;  
a network interface controller including a hardware port; and  
means for providing an application programming interface to a software-implemented switch running on said processor.
31. The electronic system of claim 30 wherein said means includes a means for registering and deregistering said application with said software-implemented switch.
32. The electronic system of claim 30 wherein said means includes a means for sending and receiving data to and from switch software-implemented switch.